

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

日本国特許庁
JAPAN PATENT OFFICE

Priority Papers



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2001年 4月11日

出願番号

Application Number:

特願2001-112354

出願人

Applicant(s):

株式会社日立製作所

U.S. Appln. Filed 8-29-01

Inventor: H. Fujii et al

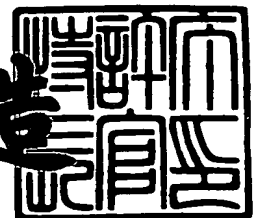
Mathingly Stanger & Malor

Docket NIT-292

2001年 6月18日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



【書類名】 特許願
 【整理番号】 NT00P1088
 【提出日】 平成13年 4月11日
 【あて先】 特許庁長官 殿
 【国際特許分類】 G06F 9/30
 G06F 9/31
 G06F 9/32
 G06F 9/45
 G06F 9/455

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

【氏名】 藤井 啓明

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

【氏名】 田中 義一

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

【氏名】 三木 良雄

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100068504

【弁理士】

【氏名又は名称】 小川 勝男

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100086656

【弁理士】

【氏名又は名称】 田中 恭助

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100094352

【弁理士】

【氏名又は名称】 佐々木 孝

【電話番号】 03-3661-0071

【手数料の表示】

【予納台帳番号】 081423

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 動的命令変換機能を有するプロセッサシステム、該プロセッサシステムを備えたコンピュータにて実行されるバイナリトランスレーションプログラム及びそのプロセッサシステムを実装した半導体デバイス

【特許請求の範囲】

【請求項 1】

異なるハードウェアプラットフォーム向けの命令バイナリコードプログラムを動的に自身の命令バイナリコードに変換しながらプログラム実行を行う動的命令変換機能を有するプロセッサシステムにおいて、前記異なるハードウェアプラットフォーム向けの命令バイナリコードプログラムから 1 命令ずつ読み出し、該 1 命令ずつをソフトウェアを介して解釈実行する処理流と、前記 1 命令ずつ必要に応じて自身の命令バイナリコードに変換し、前記 1 命令ずつ蓄積し、該蓄積された命令バイナリコード列を必要に応じて最適化する処理流を独立させ、並列に処理することを特徴とする動的命令変換機能を有するプロセッサシステム。

【請求項 2】

上記命令バイナリコード列の最適化において、繰り返し処理や手続き呼出し処理などを並列に実行出来るように複数処理流を生成するように新たな命令バイナリコード列を構成することを特徴とする請求項 1 の動的命令変換機能を有するプロセッサシステム。

【請求項 3】

上記解釈実行する処理流及び上記最適化する処理流とは別に、前記異なるハードウェアプラットフォーム向けの命令バイナリコードプログラムをキャッシュメモリに先読みする処理流を独立させ、上記解釈実行する処理流及び上記最適化する処理流と並列に処理することを特徴とする請求項 1 及び 2 の動的命令変換機能を有するプロセッサシステム。

【請求項 4】

上記最適化する処理流が、所定の単位の命令バイナリコード列最適化を完了する毎に、前記最適化した命令バイナリコード列をその最適化完了時点で上記解釈実行する処理流が実行している命令コードと入れ替え、前記解釈実行する処理流

は、異なるハードウェアプラットフォーム向けの命令バイナリコードプログラムの各命令解釈実行時に前記 1 命令に対応する最適化変換後命令バイナリコード列が存在する場合、前記最適化変換後命令バイナリコード列を実行することを特徴とする請求項 1 ～ 3 の動的命令変換機能を有するプロセッサシステム。

【請求項 5】

1 個の L S I チップに複数のマイクロプロセッサが実装されるチップマルチプロセッサとして構成され、上記複数処理流をそれぞれ異なるマイクロプロセッサで並列して同時に処理させることを特徴とする請求項 1 ～ 4 の動的命令変換機能を有するプロセッサシステム。

【請求項 6】

1 個の命令実行制御部で同時に複数の処理流を実行するように構成され、上記複数処理流を並列に実行することを特徴とする請求項 1 ～ 4 の動的命令変換機能を有するプロセッサシステム。

【請求項 7】

上記解釈実行する処理流による実行処理中の変換後命令列と、上記最適化する処理流による前記変換後命令列の最適化後の新規変換後命令列の間で変換後命令列の入れ替え処理を行う際、排他制御を実施することを特徴とする請求項 1 ～ 3 の動的命令変換機能を有するプロセッサシステム。

【請求項 8】

少なくとも 1 つの処理流から成る動的命令変換機能を有するプロセッサシステムを備え、

前記少なくとも 1 つの処理流は異種ハードウェアにて実行されるバイナリーコードプログラムを構成する複数の命令を順次先読みし、共有メモリに格納する処理流 1 と、前記共有メモリに格納された前記複数の命令を並列して同時に解釈実行する処理流 2 と、前記解釈実行された前記複数の命令の変換を行う処理流 3 から構成されることを特徴とする動的命令変換機能を有するプロセッサシステム。

【請求項 9】

請求項 8 に記載のプロセッサシステムにおいて、前記処理流 2 は、前記複数の命令の内、既に変換されている命令に対し、前記変換を実行することなく前記解

釈実行を行うことを特徴とする動的命令変換機能を有するプロセッサシステム。

【請求項 1 0】

請求項 8 に記載のプロセッサシステムにおいて、前記処理流 3 は、前記複数の命令の内、変換されていない命令に対し該命令を変換し、該変換された該命令の並べ替え又は該変換された該命令数の削減を行うことを特徴とする動的命令変換機能を有するプロセッサシステム。

【請求項 1 1】

請求項 8 に記載のプロセッサシステムにおいて、前記処理流 1、前記処理流 2 及び前記処理流 3 は互いに独立して並列に処理するように構成されることを特徴とする動的命令変換機能を有するプロセッサシステム。

【請求項 1 2】

少なくとも 1 つのマイクロプロセッサ、バス、共有メモリなどを備えた半導体デバイスにおいて、

前記少なくとも 1 つのマイクロプロセッサは、少なくとも 1 つの処理流を実行するように構成され、

前記少なくとも 1 つの処理流は異種ハードウェアにて実行されるバイナリーコードプログラムを構成する複数の命令を順次先読みし、前記共有メモリに格納する処理流 1 と、前記共有メモリに格納された前記複数の命令を並列して同時に解釈実行する処理流 2 と、前記解釈実行された前記複数の命令の変換を行う処理流 3 から構成され、

前記少なくとも 1 つのマイクロプロセッサは、並列して前記複数の命令を処理しうるように構成されることを特徴とする半導体デバイス。

【請求項 1 3】

コンピュータに複数の命令の読み出しを行う手順と、該読み出した該複数の命令の内、変換されていない命令に対し変換処理を行う手順と、該変換処理された該命令を実行する手順を並列に実行させるためのバイナリトランスレーションプログラム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、動的命令変換機能を有するプロセッサシステムに係わり、詳しくは異なるハードウェアプラットフォーム向けの命令バイナリコードプログラムを動的に自身の命令バイナリコードに変換しながらプログラム実行を行う動的命令変換機能を有するプロセッサシステム、該プロセッサシステムを備えたコンピュータにて実行されるバイナリートランスレーションプログラム及びそのプロセッサシステムを実装した半導体デバイスに関する。

【0002】

【従来の技術】

計算機システムを製造するメーカーは、当該計算機システムの性能向上を目的として同システムのCPU（中央処理装置）に従来と異なるアーキテクチャのマイクロプロセッサを適用する場合がある。

【0003】

この場合に問題となるのが、従来の計算機システムとのソフトウェア互換性である。

【0004】

基本的に、このようなアーキテクチャ変更を行った計算機システムでは、従来システムで使用していたソフトウェアが利用不能になる。

【0005】

この問題を解決する手段として、当該ソフトウェアに対するソースコードを新規システム上でコンパイラによって再コンパイルし、当該新規システム用の命令バイナリコードを生成するという方法などが紹介されている。

【0006】

しかし、当該新規システム使用者の手元にそのようなソースコードが無い場合には上記方法等が利用できないケースも多い。

【0007】

上記したようなケースでも対応可能な手法として、ソフトウェア手段によって従来計算機システムにおいて使用されていたマイクロプロセッサ向けの命令を解釈実行（interpret）する、または、当該マイクロプロセッサ向けの命

令を当該新規システムのマイクロプロセッサ向けの命令に変換 (t r a n s l a t e) して同変換後命令を直接実行するという方法がある。

【0008】

特に従来計算機システムで使用していたソフトウェアプログラムの新規システムでの処理中に動的に後者の命令変換、変換後実行を適用する方法は動的命令変換方法 (d y n a m i c b i n a r y t r a n s l a t i o n) と呼ばれ、これを実現する機能は動的命令変換機能と呼ばれる。

【0009】

これらのソフトウェア手段については、IEEE学会誌IEEE Computer 2000年3月号の40頁から45頁に収録されている “Wel come to the Opportunities of Binary Translation” という記事にて概略的に紹介されており、また、同学会誌47頁から52頁に収録されている “PA-RISC to IA-64: Transparent Execution, No Recompilation” という記事は同技術の一事例を紹介している。

【0010】

上記動的命令変換手法は、上述の計算機システムのマイクロプロセッサが変更されたようなケースへの対応としてだけでなく、あるハードウェアプラットフォームの計算機システムを利用しているユーザが、異なるハードウェアプラットフォーム上でのみ動作するソフトウェアを使用したい場合にも適用される技術である。

【0011】

また、近年になって、動的命令変換機能を積極的にアーキテクチャに取り込んだ新規マイクロプロセッサの提案が相次ぎ、注目されている。

【0012】

具体的な事例としては、IEEE学会誌IEEE Computer 2000年3月号の54頁から59頁に収録の “Dynamic and Transparent Binary Translation” で開示されているIBM社のBOA (B i n a r y - t r a n s l a t i o n O p t i m i z e d

Architecture) や、Cahners MICROPROCESSOR REPORT Volume 14, Archive 2 の1頁および9頁から18頁に収録の“TRANSMETA BREAKS X86 LOW-POWER BARRIER -VLIW Chips Use Hardware-Assisted x86 Emulation”で開示されているTransmeta社のCrusoeなどが挙げられる。

【0013】

図2に上記従来技術の動的命令変換機能を含んだ、異種ハードウェアプラットフォーム向け命令バイナリコードプログラム（以降、元命令列と略称する場合がある）実行機構の構成を示す。

【0014】

図2において、201は異種ハードウェアプラットフォーム向け命令の解釈実行部、202は本実行機構の処理全体を制御する実行制御部、203は異種ハードウェアプラットフォーム向け命令列から本実行機構を有するハードウェアプラットフォームの命令列（以降、変換後命令列と略称する場合がある）を動的に生成する動的命令変換部、204はオペレーティングシステム（OS）が関係する処理部分等のプログラム中の特殊処理部を本実行機構を有するハードウェアプラットフォームの機能を用いてエミュレーションする特殊処理エミュレーション部、そして、205は本実行機構を有するプラットフォームOS及びハードウェアである。

【0015】

本実行機構による異種ハードウェアプラットフォーム向け命令バイナリコードプログラム処理がプラットフォームOS及びハードウェア205上で起動されると、実行制御部202が処理を開始する。実行制御部202は、プログラム処理中に適時、解釈実行部201、動的命令変換部203、及び特殊処理エミュレーション部204に処理を依頼する。特殊処理エミュレーション部204は、プラットフォームOS及びハードウェア205の機能を直接使用して依頼された処理を遂行する。

【0016】

次に図 2 に係わるより詳細な処理フローに関して、図 3 を用いて説明する。

【0017】

図 2 の実行機構が処理を開始すると、実行制御部 2 0 2 が動作を始め、処理 3 0 1 のとおり、元命令列アドレスに基づき、元命令列中の命令が参照され、当該命令に対する実行回数計測カウンタがインクリメントされる。実行回数計測カウンタは、元命令列管理表などのソフトウェアデータ構造体中に含まれる。

【0018】

次に、処理 3 0 2 において、当該元命令列管理表が参照され、当該命令に対応する変換後命令列の有無がチェックされる。変換後命令列が存在する場合、変換後命令列領域 3 0 8 上の該当する変換後命令列ブロック 3 0 6 を元命令列管理表を参照して特定し、これを直接実行した後処理 3 0 1 に戻る。処理 3 0 2 において、変換後命令列が存在しない場合、当該命令の実行回数が検査される。同実行回数があらかじめ定められた閾値を超える場合、処理 3 0 5 が起動され、閾値以下の場合、処理 3 0 4 が起動される。処理 3 0 4 開始にあたって、実行制御部 2 0 2 は、解釈実行部 2 0 1 を呼び出す。解釈実行部 2 0 1 は、元命令列を順次参照し、各命令を解釈し、それぞれの動作に相当する処理をあらかじめ用意されたソフトウェア処理手順にしたがって実現する。

【0019】

先述したとおり、当該命令がオペレーティングシステム（OS）が関係する処理部分等のプログラム中の特殊処理部に該当する場合には、解釈実行部 2 0 1 は、実行制御部 2 0 2 にその旨を報告する。実行制御部 2 0 2 は、特殊処理エミュレーション部 2 0 4 を起動し、特殊処理エミュレーション部 2 0 4 は、プラットフォーム OS 及びハードウェア 2 0 5 の機能を用いて、当該処理を実施する。特殊処理が完了すると、制御は、特殊処理エミュレーション部 2 0 4 から実行制御部 2 0 2 を介して、解釈実行部 2 0 1 に戻る。解釈実行部 2 0 1 は、元命令列に分岐命令が出てくるまで上記処理を繰り返した後、制御を実行制御部 2 0 2 の処理 3 0 1 に戻す。

【0020】

一方、処理 3 0 5 開始にあたって、実行制御部 2 0 2 は、動的命令変換部 2 0

3を呼び出す。動的命令変換部203は、分岐命令で区切られる一連の元命令列（ブロック）中の各命令を本実行機構を有するハードウェアプラットフォームの命令列に置換え、生成した変換後命令列を必要に応じて最適化した後、同変換後命令列を変換後命令列ブロック306として変換後命令列領域308上に保存する。

【0021】

その後、動的命令変換部203は、制御を実行制御部202に戻し、実行制御部202は、当該新規に生成された変換後命令列ブロック306を直接実行した後処理301に戻る。実行制御部202は、以上の処理をプログラム終了まで繰り返す。なお、以上で説明した処理の分担は一例であり、異なる処理分担の例もあり得る。

【0022】

上記処理フローは、1つの処理流で実現されている。このため、動的命令変換部203における命令変換及び最適化処理は元命令列実行処理にとってオーバヘッドとなり、元命令列処理性能を低下させる。

【0023】

また、上記IBM社のBOAやTransmeta社のCrusoeは、基本アーキテクチャにVLIW（Very long Instruction Word）方式を採用して、変換後命令列の命令レベル並列処理による高速処理、プロセッサ自体の高速周波数での動作、低消費電力化を実現することを狙っているが、上記の動的命令変換部203における命令変換及び最適化処理のオーバヘッドの削減は必ずしも十分でなく、更なる削減が望まれる。また、LSIテクノロジーの将来動向を考慮すると、プロセッサの高速周波数動作や低消費電力化という目的に対しても、VLIW方式が最良の方式であるとは必ずしも言えない。

【0024】

【発明が解決しようとする課題】

以下3点が本発明によって解決しようとする課題である。

(1) 上記動的命令変換部203における命令変換及び最適化処理のオーバヘッドを削減する。

(2) 異種プロセッサ用プログラムの先読み処理を他の解釈実行処理、命令変換・最適化処理と並列に行い、プログラム処理の性能を向上する。

(3) 変換後命令列の高速処理、プロセッサの高速周波数動作及びプロセッサ低消費電力化をV L I W方式よりも効果的に実現する。

【 0 0 2 5 】

【課題を解決するための手段】

上記課題を解決するために、本発明の動的命令変換機能を有するプロセッサシステムは、異種ハードウェアプラットフォーム向けの命令バイナリコードプログラムを動的に自身の命令バイナリコードに変換しながらプログラム実行を行う際、当該異種ハードウェアプラットフォーム向けの命令バイナリコードプログラムから1命令ずつ読み出し、該1命令ずつをソフトウェアを介して解釈実行する処理流と、前記1命令ずつ必要に応じて自身の命令バイナリコードに変換し、前記1命令ずつ蓄積し、蓄積された命令バイナリコード列を必要に応じて最適化する処理流を独立させてこれらを並列に処理することを特徴とする。

【 0 0 2 6 】

さらに、上記命令バイナリコード列の最適化において、繰り返し処理や手続き呼出し処理などを並列に実行出来るように複数処理流を生成するように新たな命令バイナリコード列を構成し、上記解釈実行する処理流及び上記最適化する処理流とは別に、前記異なるハードウェアプラットフォーム向けの命令バイナリコードプログラムをキャッシュメモリに先読みする処理流を独立させ、上記解釈実行する処理流及び上記最適化する処理流と並列に処理することを特徴とする。

【 0 0 2 7 】

また、上記最適化する処理流が所定の単位の命令バイナリコード列最適化を完了する毎に、前記最適化した命令バイナリコード列をその最適化完了時点で上記解釈実行する処理流が実行している命令コードと入れ替え、前記解釈実行する処理流は異なるハードウェアプラットフォーム向けの命令バイナリコードプログラムの各命令解釈実行時に前記1命令に対応する最適化変換後命令バイナリコード列が存在する場合、前記最適化変換後命令バイナリコード列を実行する機構を有し、さらには、上記複数処理流を効率良く並列処理するために、1個のL S Iチ

ップに複数のマイクロプロセッサを実装するチップマルチプロセッサ形式、または、1個の命令実行制御部で同時に複数の処理流を実行する形式で構成される。

【0028】

さらに、本発明は少なくとも1つの処理流から成る動的命令変換機能を有するプロセッサシステムを備え、前記少なくとも1つの処理流は異種ハードウェアにて実行されるバイナリーコードプログラムを構成する複数の命令を順次先読みし、共有メモリに格納する処理流1と、前記共有メモリに格納された前記複数の命令を並列して同時に解釈実行する処理流2と、前記解釈実行された前記複数の命令の変換を行う処理流3から構成されることを特徴とする動的命令変換機能を有するプロセッサシステムを提供することである。

【0029】

さらに、本発明は少なくとも1つのマイクロプロセッサ、バス、共有メモリなどを備えた半導体デバイスにおいて、前記少なくとも1つのマイクロプロセッサは、少なくとも1つの処理流を実行するように構成され、前記少なくとも1つの処理流は異種ハードウェアにて実行されるバイナリーコードプログラムを構成する複数の命令を順次先読みし、前記共有メモリに格納する処理流1と、前記共有メモリに格納された前記複数の命令を並列して同時に解釈実行する処理流2と、前記解釈実行された前記複数の命令の変換を行う処理流3から構成され、前記少なくとも1つのマイクロプロセッサは、並列して前記複数の命令を処理しうるように構成されることを特徴とする半導体デバイスを提供することにある。

【0030】

しかも、本発明はコンピュータに複数の命令の読み出しを行う手順と、該読み出した該複数の命令の内、変換されていない命令に対し変換処理を行う手順と、該変換処理された該命令を実行する手順を並列に実行させるためのバイナリトランスレーションプログラムを提供することにある。

【0031】

【発明の実施の形態】

本発明の実施の形態を図を用いながら説明する。

【0032】

図4に本発明に係わる動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構の構成を示す。

【0033】

本実行機構は、実行制御部401、元命令列解釈実行部402、命令変換・最適化処理部403、元命令列先読み処理部404の各処理部と、主記憶408上のデータ構造体である元命令列407、複数の変換後命令列410を含む変換後命令列領域409、命令対応表411等で構成される。

【0034】

命令対応表411は例えば図5のような形で構成される。

【0035】

命令対応表411中の各エントリー506は、元命令列中の各命令に対応して用意され、例えば当該命令の元命令列中での先頭からの相対アドレスを用いて一意に識別される。

【0036】

各エントリー506は、変換後命令列有無ビットフィールド501、命令実行回数フィールド502、その他プロファイル情報フィールド503、対応変換後命令列先頭アドレスフィールド504、実行中ビットフィールド505などで構成される。

【0037】

変換後命令列有無ビットフィールド501は、当該エントリー506に対応する元命令に対する変換後命令列410が存在するか否かを表示する。当該変換後命令列有無ビットフィールド501が、当該エントリー506に対応した元命令に対する変換後命令列410が存在することを示す内容となっている（例えば「1」を表示）場合、対応変換後命令列先頭アドレスフィールド504の値は、当該変換後命令列410の主記憶408における先頭アドレスである。

【0038】

逆に、当該変換後命令列有無ビットフィールド501が、当該エントリー506に対応した元命令に対する変換後命令列410が存在しないことを示す内容となっている（例えば「0」を表示）場合、対応変換後命令列先頭アドレスフィー

ルド504の値は無効である。

【0039】

また、命令実行回数フィールド502は、当該エントリー506に対応する元命令の実行回数を表示する。当該命令実行回数フィールド502の値が所定の閾値を越えている場合、当該エントリー506に対応した元命令が命令変換・最適化処理部403での命令変換及び最適化の対象となる。

【0040】

さらに、その他プロファイル情報フィールド503は、当該エントリー506に対応した元命令の実行時に発生した事象をプロファイルとして記録しておくフィールドである。

【0041】

例えば当該元命令が条件分岐命令であった場合、分岐条件成立／不成立といった情報が当該その他プロファイル情報フィールド503に記録される。また、命令変換・最適化処理部403での命令変換及び最適化に有益なプロファイル情報なども当該その他プロファイル情報フィールド503に記録される。実行中ビットフィールド505は、当該エントリー506に対応する元命令に対する変換後命令列410が存在する場合、又は元命令列解釈実行部402が当該変換後命令列410を実行中である場合にその旨を示す値（例えば「1」）を表示する。

【0042】

これ以外の場合、実行中ビットフィールド505は、無効値（例えば「0」）を表示している。各フィールドの初期値については、変換後命令列有無ビットフィールド501、実行中ビットフィールド505が無効値（例えば「0」）であり、命令実行回数フィールド502の値は「0」、その他プロファイル情報フィールド503も無効値である。

【0043】

次に、図4に戻って各構成要素の概略動作を説明する。

【0044】

異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行が開始されると、まず実行制御部401が元命令列解釈実行部402、命令変換・

最適化処理部403，元命令列先読み処理部404のそれぞれに対応する3つの独立した処理流を生成する。

【0045】

元命令列先読み処理部404に対応する処理流は、実行する元命令列407の先読み処理を行う。

【0046】

先読みされた元命令列は、元命令列のコピー405という形でキャッシュメモリ406上に存在することになる。元命令列解釈実行部402及び命令変換・最適化処理部403が元命令列407をアクセスする際には既にキャッシュメモリ406上に存在する当該元命令列のコピー405をアクセスできる。

【0047】

元命令列先読み処理部404の先読みした元命令が分岐命令であった場合、元命令列先読み処理部404は一旦分岐双方向の命令列を一定数先読みしておき、実際に当該分岐命令が元命令列解釈実行部402にて処理されるのを待ち、その処理終了後当該分岐命令に対応する命令対応表411中のその他プロファイル情報フィールド503の値を参照して正しい分岐方向を特定し、以降その方向で元命令列先読みを継続する。

【0048】

元命令列解釈実行部402に対応する処理流は、元命令列中の各命令の解釈実行または当該命令に対応する変換後命令列410が存在する場合には当該変換後命令列410の直接実行を行う。当該命令を解釈実行するかあるいは当該命令に対応する変換後命令列410を直接実行するかは、命令対応表411の変換後命令列有無ビットフィールド501の値を確認して判断する。

【0049】

当該命令に対応する変換後命令列有無ビットフィールド501の値が対応する元命令に対する変換後命令列410が存在しないことを示す内容となっている（例えば「0」を表示）場合、元命令列解釈実行部402は当該命令の解釈実行を行う。

【0050】

逆に、当該変換後命令列有無ビットフィールド501の値が対応する元命令に対する変換後命令列410が存在することを示す内容となっている（例えば「1」を表示）場合、元命令列解釈実行部402は当該命令に対応する対応変換後命令列先頭アドレスフィールド504の値から対応する変換後命令列410を特定し、当該変換後命令列410を直接実行する。

【0051】

この際、元命令列解釈実行部402は、当該変換後命令列410の直接実行に先だって当該命令に対応する実行中ビットフィールド505の値を有効（例えば「1」）にし、さらに、当該変換後命令列410の直接実行終了時に当該実行中ビットフィールド505の値を無効（例えば「0」）にする。

【0052】

また、元命令列解釈実行部402は、各元命令に対応する解釈実行または変換後命令列直接実行を実施する毎に当該元命令の実行回数を当該命令に対応する命令実行回数フィールド502に書き込み、さらに実行プロファイル情報を当該命令に対応するその他プロファイル情報フィールド503に書き込む。

【0053】

命令変換・最適化処理部403に対応する処理流は、元命令列の当該プロセッサシステム自身の命令列への変換及び当該変換後命令列の最適化を行う。

【0054】

命令変換・最適化処理部403は、命令対応表411中の各元命令に対応する命令実行回数フィールド502を参照し、この値が所定の閾値を越えている場合に、当該元命令を当該プロセッサシステム自身の命令列に変換して変換後命令列410を主記憶408上の変換後命令列領域409に作成し、さらに、前後の元命令に対応する変換後命令列410が存在する場合、これらと合わせて最適化を行い、新たな最適化変換後命令列410を作成する。

【0055】

最適化に際しては、前後の元命令含めて対応する命令対応表411中のその他プロファイル情報フィールド503の値を参照し、最適化のためのヒント情報として使用する。

【0056】

変換後命令列410を作成した命令変換・最適化処理部403は、対応する元命令に対する命令対応表411中の変換後命令列有無ビットフィールド501の値を検査し、これが無効（例えば「0」）を示している場合には、当該変換後命令列有無ビットフィールド501の値を有効（例えば「1」）を示す値に書き換え、対応する対応変換後命令列先頭アドレスフィールド504に生成した変換後命令列410の主記憶408上での先頭アドレスを書き込む。

【0057】

逆に、当該変換後命令列有無ビットフィールド501の値が有効（例えば「1」）を示す値であった場合には、さらに対応する実行中ビットフィールド505を検査し、これが無効（例えば「0」）を示している場合には、対応する対応変換後命令列先頭アドレスフィールド504に生成した変換後命令列410の主記憶408上での先頭アドレスを書き込み、元々当該対応変換後命令列先頭アドレスフィールド504で示されていた変換後命令列410のメモリ領域を解放する。

【0058】

この時、当該実行中ビットフィールド505の値が有効（例えば「1」）を示す値であった場合には、当該実行中ビットフィールド505の値が無効（例えば「0」）を示す値になるのを待ってから対応する対応変換後命令列先頭アドレスフィールド504に生成した変換後命令列410の主記憶408上での先頭アドレスを書き込み、元々当該対応変換後命令列先頭アドレスフィールド504で示されていた変換後命令列410のメモリ領域を解放する。

【0059】

続いて、図1を用いて本発明に係わる動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構の処理フローを詳細に説明する。

【0060】

処理101で動的命令変換部による異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行が始まる。続いて処理102で処理流が3つに

分割される。

【0061】

ここで生成された元命令列先読み処理流103，元命令列解釈実行処理流104，命令変換・最適化処理流105の3処理流は並列に動作する。

【0062】

以降、個々の処理流の処理フローについて説明する。まず、元命令列先読み処理流103の処理フローを説明する。処理106で元命令列先読み処理が開始される。

【0063】

次に処理107で元命令列の実行順に元命令が先読みされる。処理108では先読みした元命令の種類を解釈する。この元命令が分岐命令であるかどうかは処理109で判定され、分岐命令であれば処理110に、そうでなければ処理113に移る。処理110では、当該分岐命令の分岐双方向に元命令列の実行順に元命令を先読みする。

【0064】

次に処理111では、当該分岐命令に対応する命令対応表411中のその他プロファイル情報フィールド503を参照して正しい分岐方向を得る。処理112では、正しい分岐方向パスでの先読みをした元命令の種類を解釈する。以降処理109に戻って処理が繰り返される。

【0065】

一方、処理113では、次に先読みをすべき領域が元命令列プログラムの領域外であるかどうかを判定する。領域外であれば、処理115に移り、元命令列先読み処理を終了する。領域外でなければ、処理114に移る。処理114では、元命令列解釈実行処理流104が終了しているかどうか判定される。処理が終了していれば処理115に移り、元命令列先読み処理を終了する。一方、終了していなければ処理107に移り以降処理を繰り返す。

【0066】

次に、元命令列解釈実行処理流104の処理フローを説明する。処理116で元命令列解釈実行処理が始まる。

【0067】

処理117では、元命令列の実行順で次の（一番はじめの時は先頭の）元命令に対応する命令対応表411中の変換後命令列有無ビットフィールド501を参照し、当該元命令に対する変換後命令列410が存在するか否かを判定する。変換後命令列410が存在すれば、処理123に移り、存在しなければ処理119に移る。処理119では当該元命令を解釈実行し、処理122に移る。処理123では、当該変換後命令列410の実行に先だって当該元命令に対応する命令対応表411中の実行中ビットフィールド505に当該変換後命令列410の実行中を示す値（例えば「1」）を書き込む。

【0068】

次に処理118で当該変換後命令列410の直接実行が開始される。当該直接実行処理中、処理120でマルチスレッド処理が開始された場合、処理121で当該マルチスレッド処理が実施される。当該変換後命令列410が最後まで実行されると、処理139で当該直接実行処理が終了したことが判定され、処理124に移る。処理124では、当該元命令に対応する命令対応表411中の実行中ビットフィールド505に当該変換後命令列410を実行中で無いことを示す値（例えば「0」）を書き込む。

【0069】

次に、処理122では、当該元命令に対応する命令対応表411中の命令実行回数フィールド502、その他プロファイル情報フィールド503に実行結果を反映する。続く処理125では、次の元命令が存在するかどうかを判定し、存在しなければ処理126に移り、元命令列解釈実行処理を終了する。次の元命令が存在する場合には、処理117に戻り、以降処理を繰り返す。

【0070】

続いて、命令変換・最適化処理流105の処理フローを説明する。処理127で命令変換・最適化処理が開始される。

【0071】

処理128では、命令対応表411中の命令実行回数フィールド502、その他プロファイル情報フィールド503を順次参照する。処理129では、当該命

令実行回数フィールド502の値が所定の閾値を越えているかどうか判定され、閾値を越えている場合には、処理130に移り、閾値を越えていない場合には処理128に戻る。

【0072】

処理130では、当該命令実行回数フィールド502の値が所定の閾値を越えている命令対応表411中のエントリー506に対応する元命令の命令変換を実施し、変換後命令列410を主記憶408における変換後命令列領域に生成する。

【0073】

なお、当該変換後命令列410生成時には、当該元命令に対応する命令対応表411中のその他プロファイル情報フィールド503の値を変換後命令列最適化のための情報として使用する。

【0074】

続いて、処理131で、当該元命令の前後の元命令に対応する変換後命令列410が存在した場合、それらの変換後命令列410を合わせて再度最適化処理を実施する。

【0075】

最適化処理において、処理132でマルチスレッド処理化した方がプログラムの処理効率が上がると判定された場合、処理133でマルチスレッド処理化を実施する。

【0076】

続いて、処理134で当該元命令に対応する命令対応表411中の変換後命令列有無ビットフィールド501に変換後命令列410が存在することを示す値（例えば「1」）を書き込み、さらに、同エントリー506の対応変換後命令列先頭アドレスフィールド504に当該変換後命令列410の主記憶408における先頭アドレスを書き込む。

【0077】

処理135では、当該元命令に対応する命令対応表411中の実行中ビットフィールド505を参照し、対応する旧変換後命令列が実行中であるかどうかを判

定する。

【0078】

実行中であれば実行終了するまで待つ。実行中でなければ、処理136にて当該旧変換後命令列410のメモリ領域を解放し、廃棄する。

【0079】

次に処理137では、元命令列解釈実行処理が終了しているかどうかを判定し、終了していれば処理138に移って命令変換・最適化処理を終了する。元命令列解釈実行処理が終了していなければ、処理128に戻って以降処理を繰り返す。

【0080】

以上が本発明に係わる動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構の処理フローである。

【0081】

ここで、前述される最適化処理とは、変換後命令の並べ替え及び変換後命令数の削減等により命令コードをコンパイラ等のソフトウェアを通して実行コード化した時の実行スピード向上を目指した処理に相当する。

【0082】

さらに、マルチスレッド処理とは、従来はプログラムの各命令を順次実行するのに対し、各命令を各々のマイクロプロセッサにて並列に同時実行することによる処理能率向上を目指した処理に相当する。

【0083】

次に、図7、図8を用いて、元命令列先読み処理流103、元命令列解釈実行処理流104、および命令変換・最適化処理流105の相関関係を共有データ構造へのアクセスに着目して説明する。

【0084】

図7は、キャッシュメモリ406上に生成される元命令列のコピー405へのアクセスに関する各処理流の相関関係を示す。元命令列のコピー405は、元命令列先読み処理流103の処理107及び処理110における元命令先読みによってキャッシュメモリ406上に生成される。この元命令列のコピー405は、

元命令列解釈実行処理流 1 0 4 の処理 1 1 9, および命令変換・最適化処理流 1 0 5 の処理 1 3 0 における元命令読み出し時にアクセスされる。

【 0 0 8 5 】

図 8 は、主記憶 4 0 8 上に生成される命令対応表 4 1 1 のエントリー 5 0 6 の変換後命令列有無ビットフィールド 5 0 1, 命令実行回数フィールド 5 0 2, その他プロファイル情報フィールド 5 0 3, 対応変換後命令列先頭アドレスフィールド 5 0 4, 実行中ビットフィールド 5 0 5 の各フィールドと、主記憶 4 0 8 内の変換後命令列領域 4 0 9 上に生成される変換後命令列 4 1 0 へのアクセスに関する各処理流の相関関係を示す。

【 0 0 8 6 】

まず第 1 に変換後命令列有無ビットフィールド 5 0 1 は、命令変換・最適化処理流 1 0 5 の処理 1 3 4 によって値が更新され、元命令列解釈実行処理流 1 0 4 の処理 1 1 7 で参照される。

【 0 0 8 7 】

次に命令実行回数フィールド 5 0 2 は、元命令列解釈実行処理流 1 0 4 の処理 1 2 2 によって値が更新され、命令変換・最適化処理流 1 0 5 の処理 1 2 8 から処理 1 2 9 までの処理群 8 0 2 で参照される。その他プロファイル情報フィールド 5 0 3 は、元命令列解釈実行処理流 1 0 4 の処理 1 2 2 によって値が更新され、元命令列先読み処理流 1 0 3 の処理 1 1 1 及び命令変換・最適化処理流 1 0 5 の処理 1 3 0 から処理 1 3 3 までの処理群 8 0 1 によって参照される。

【 0 0 8 8 】

対応変換後命令列先頭アドレスフィールド 5 0 4 は、命令変換・最適化処理流 1 0 5 の処理 1 3 4 によって値が更新され、元命令列解釈実行処理流 1 0 4 の処理 1 1 8 から処理 1 3 9 までの処理群 8 0 3 で参照される。

【 0 0 8 9 】

さらに、実行中ビットフィールド 5 0 5 は、元命令列解釈実行処理流 1 0 4 の処理 1 2 3 及び処理 1 2 4 によって値が更新され、命令変換・最適化処理流 1 0 5 の処理 1 3 5 で参照される。

【 0 0 9 0 】

最後に変換後命令列410は、命令変換・最適化処理流105の処理130から処理133までの処理群801によって生成され、元命令列解釈実行処理流104の処理118から処理139までの処理群803で参照される。

【0091】

ここで元命令列解釈実行処理流104による実行処理中の変換後命令列と命令変換・最適化処理流105による変換後命令列の最適化後の新規変換後命令列との間で変換後命令列の入れ替え処理を行う際、排他制御（すなわち、主記憶に内在する共有メモリを処理流104及び処理流105などが利用する場合、一方の処理流が使用している時、他方の処理流は共有メモリの使用から排除される）が実施される。

【0092】

ここまででは、本発明に係わる動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構の処理方式について説明した。

【0093】

引き続き以降では、以上のような処理を実行可能なハードウェアプラットフォームについて記述する。

【0094】

まず図6は、チップマルチプロセッサ605の構成例を示す。

【0095】

本ハードウェアプラットフォームの具体例については、Proceedings of Eighth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS VIII)の58ページから69ページに掲載されている“Data Speculation Support for a Chip Multiprocessor”というタイトルの論文に開示されている。

【0096】

チップマルチプロセッサ605中には、複数のマイクロプロセッサ601と、

これらのマイクロプロセッサ601を互いに接続する相互結合網602，当該相互結合網602に接続され、該複数マイクロプロセッサ601で共有される共有キャッシュ603、および、主記憶インタフェース604などが含まれる。

【0097】

本発明の処理方式の下で発生する複数処理流はそれぞれスレッドと呼ばれ、当該チップマルチプロセッサ605中の該複数マイクロプロセッサ601に各々のスレッドが個別に割り当てること、コンパイラ等のソフトウェアを介して上記複数処理流の並列処理が実現される。

【0098】

次に、図9は、同時複数スレッド実行プロセッサ909の構成例を示す。

【0099】

本ハードウェアプラットフォームの具体例については、IEEE Micro誌1997年9-10月号12ページから19ページに掲載されている“Simultaneous Multithreading: A Platform for Next-Generation Processors”というタイトルの論文に開示されている。

【0100】

同時複数スレッド実行プロセッサ909は、命令キャッシュ901，複数の命令フェッチ部902（命令フェッチ部902-1から命令フェッチ部902-n），命令選択合成部903，命令デコード部904，命令実行部905，複数のレジスタセット906（レジスタセット906-1からレジスタセット906-n），主記憶インタフェース907，データキャッシュ908などからなる。

【0101】

このうち、命令キャッシュ901，命令デコード部904，命令実行部905，主記憶インタフェース907，データキャッシュ908は通常のマイクロプロセッサのものと基本的に同じである。

【0102】

特徴的なのは、複数の命令フェッチ部902（命令フェッチ部902-1から命令フェッチ部902-n），命令選択合成部903，及び複数のレジスタセッ

ト906（レジスタセット906-1からレジスタセット906-n）の各部である。複数の命令フェッチ部902（命令フェッチ部902-1から命令フェッチ部902-n），及び複数のレジスタセット906（レジスタセット906-1からレジスタセット906-n）は，それぞれ本発明の同時複数スレッド実行プロセッサ909で同時に処理するスレッド毎に1つつ割り当てられる。

【0103】

命令選択合成部903は、各時点での各スレッドの処理状況に応じて命令を取り出す命令フェッチ部902を制限し、制限した当該命令フェッチ部902が持つ実行可能命令候補の中から同時実行可能な組合せで複数命令を選択し、命令デコード部904に引き渡す。

【0104】

本発明の処理方式の下で発生する複数処理流をそれぞれスレッドとして、当該同時複数スレッド実行プロセッサ909中の該命令フェッチ部902（命令フェッチ部902-1から命令フェッチ部902-n），及び該レジスタセット906（レジスタセット906-1からレジスタセット906-n）の組に個別に割り当てることで、上記複数処理流の並列処理が実現される。

【0105】

以上が、本発明に係わる実施の形態である。

【0106】

【発明の効果】

本発明によって、異種プロセッサ用プログラムを動的に自身の命令列に変換しながらプログラム実行を行う際、命令変換及び最適化処理のオーバーヘッドが削減できる。

【0107】

さらには、異種プロセッサ用プログラムの先読み処理を他の解釈実行処理及び命令変換・最適化処理と並列に行うことで同プログラム処理の性能を向上できる。

【0108】

また、特に、チップマルチプロセッサ方式との組み合わせによって、変換後命

命令列の高速処理，プロセッサの高速周波数動作，及び低消費電力化を実現できる。

【図面の簡単な説明】

【図 1】

本発明に係わる動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構の処理フローを示す図。

【図 2】

従来技術での動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構の構成を示す図。

【図 3】

従来技術での動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構の処理フローを示す図。

【図 4】

本発明に係わる動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構の構成を示す図。

【図 5】

本発明に係わる動的命令変換機能を含んだ異種ハードウェアプラットフォーム向け命令バイナリコードプログラム実行機構が使用する命令対応表の構成を示す図。

【図 6】

従来技術のチップマルチプロセッサの構成例を示す図。

【図 7】

本発明に係わるキャッシュメモリ上元命令列のコピーを介した各処理流間の相互関係を示す図。

【図 8】

本発明に係わる主記憶上命令対応表及び変換後命令列領域を介した各処理流間の相互関係を示す図。

【図 9】

従来技術の同時複数スレッド実行プロセッサの構成例を示す図。

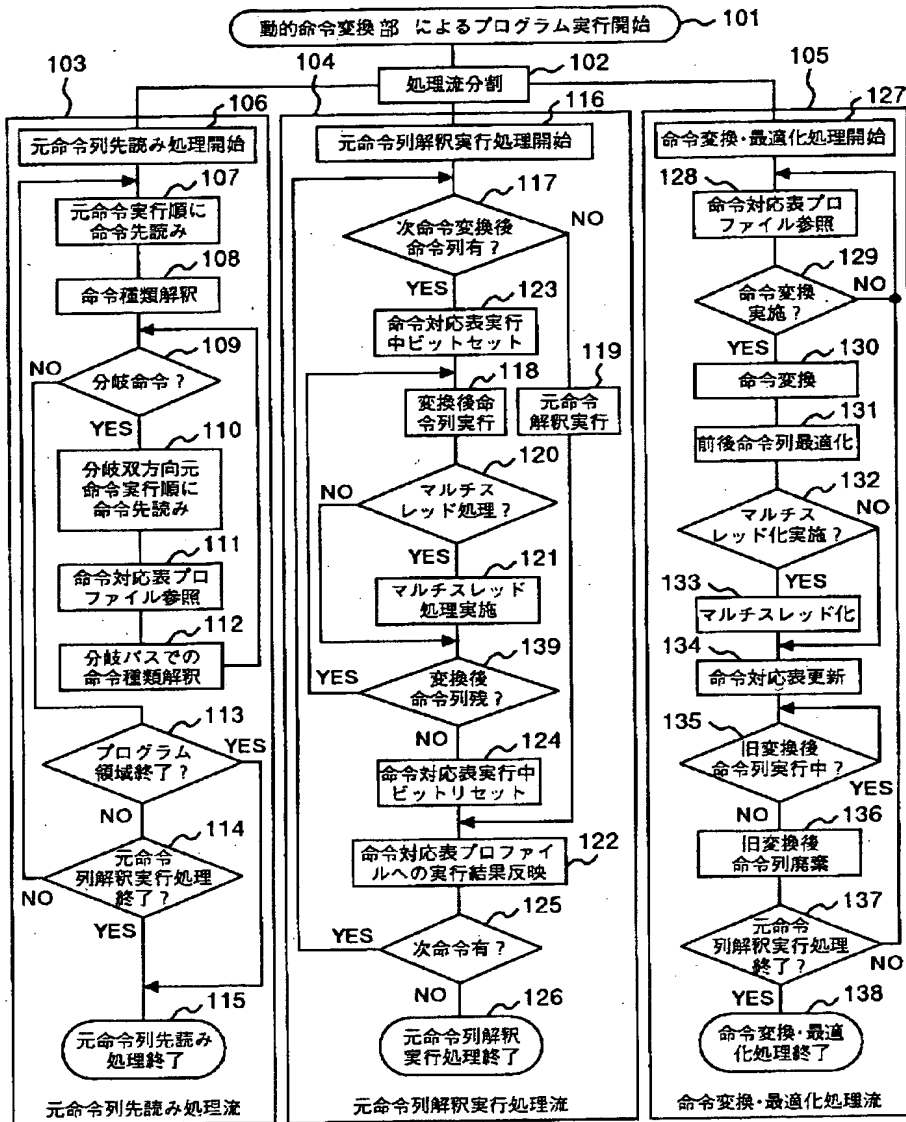
【符号の説明】

103…元命令列先読み処理流, 104…元命令列解釈実行処理流, 105…命令変換・最適化処理流, 201…解釈実行部, 202…実行制御部, 203…動的命令変換部, 204…特殊処理エミュレーション部, 205…プラットフォームOS及びハードウェア, 306…変換後命令列ブロック, 308…変換後命令列領域, 401…実行制御部, 402…元命令列解釈実行部, 403…命令変換・最適化処理部, 404…元命令列先読み処理部, 405…元命令列のコピー, 406…キャッシュメモリ, 407…元命令列, 408…主記憶, 409…変換後命令列領域, 410…変換後命令列, 411…命令対応表, 501…変換後命令列有無ビットフィールド, 502…命令実行回数フィールド, 503…その他プロファイル情報フィールド, 504…対応変換後命令列先頭アドレスフィールド, 505…実行中ビットフィールド, 506…命令対応表エントリ, 601…マイクロプロセッサ, 602…相互結合網, 603…共有キャッシュ, 604…主記憶インタフェース, 605…チップマルチプロセッサ, 901…命令キャッシュ, 902-1～902-n…命令フェッチ部1～命令フェッチ部N, 903…命令選択合成部, 904…命令デコード部, 905…命令実行部, 906-1～906-n…レジスタセット1～レジスタセットN, 907…主記憶インタフェース, 908…データキャッシュ, 909…同時複数スレッド実行プロセッサ。

【書類名】 図面

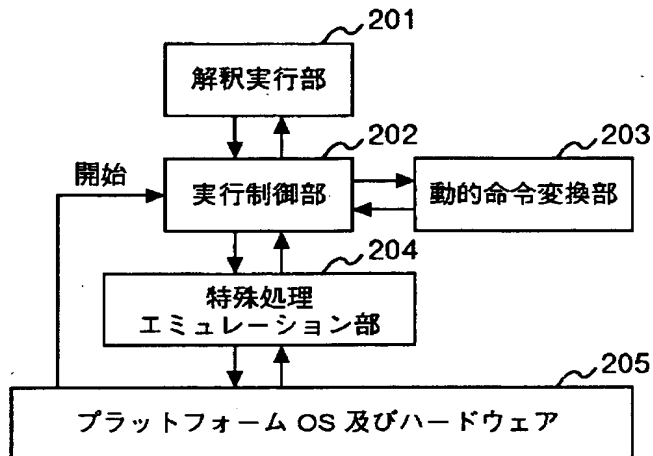
【図 1】

図 1



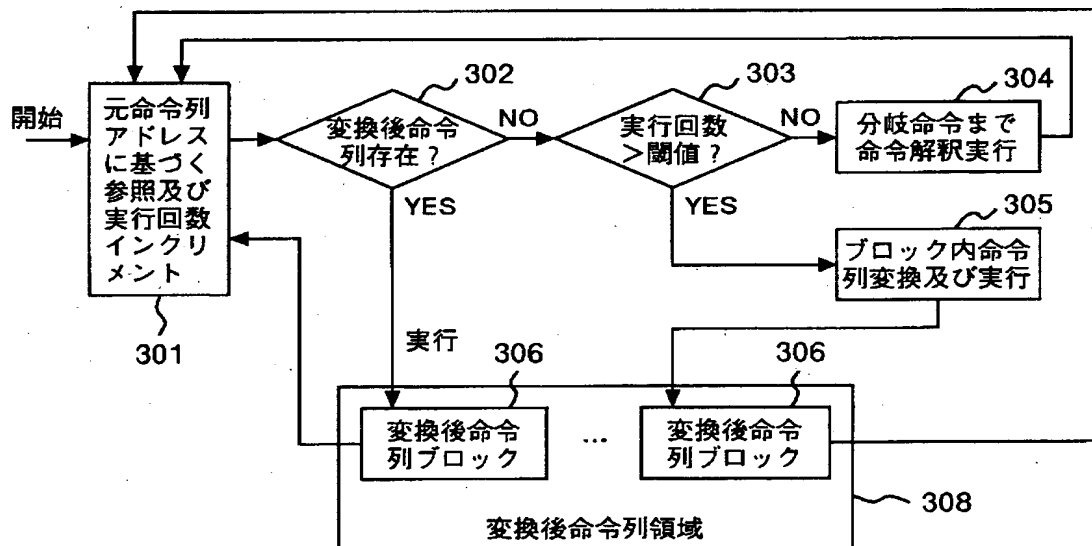
【図 2】

図 2



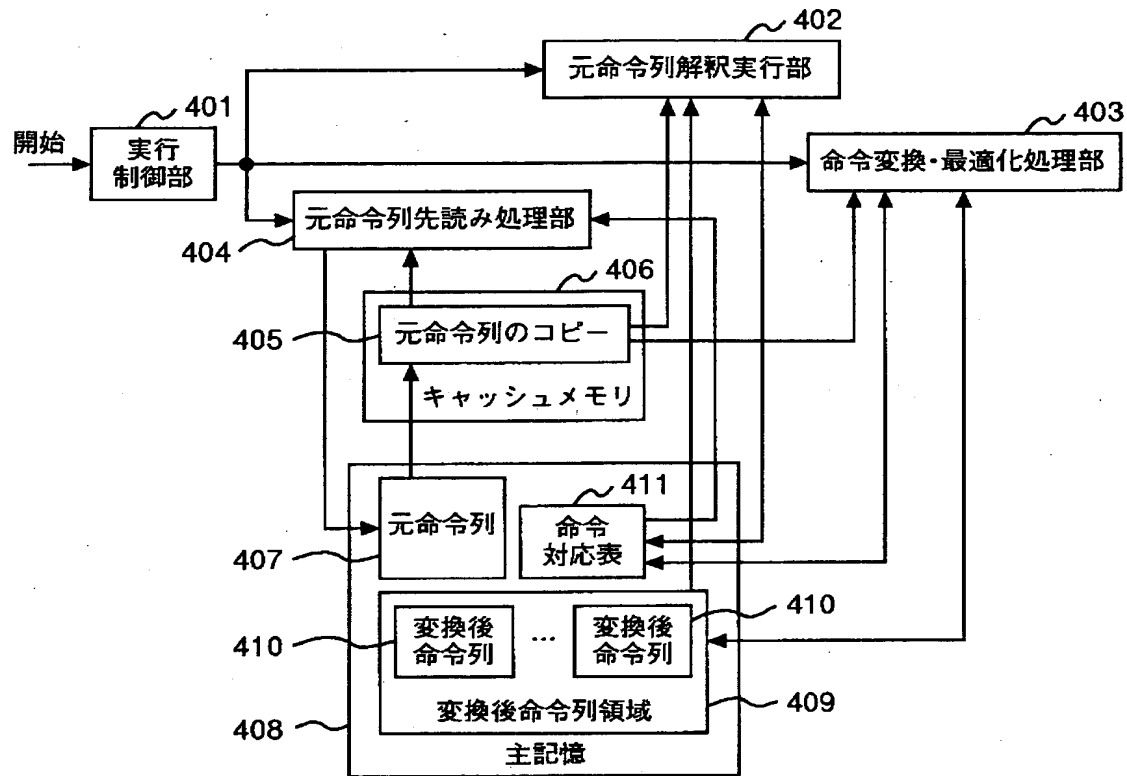
【図 3】

図 3



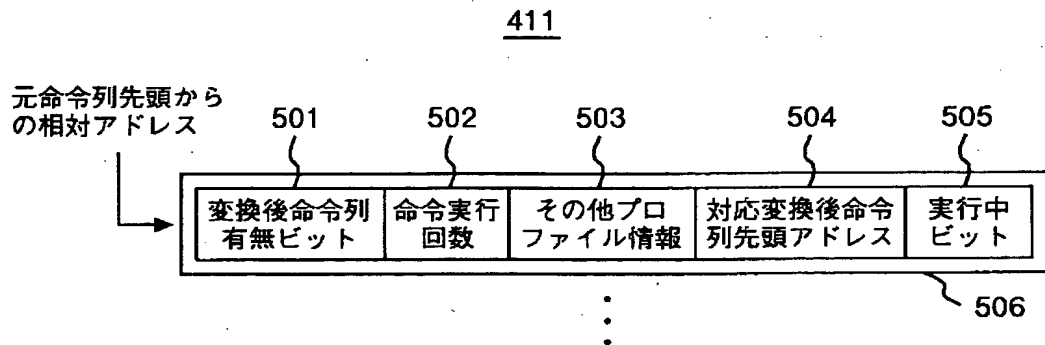
【図 4】

図 4



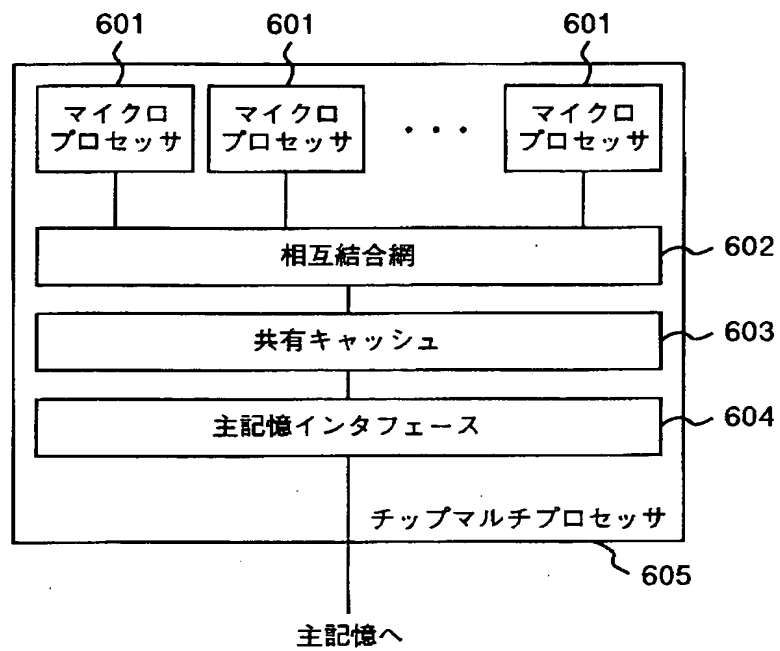
【図 5】

図 5



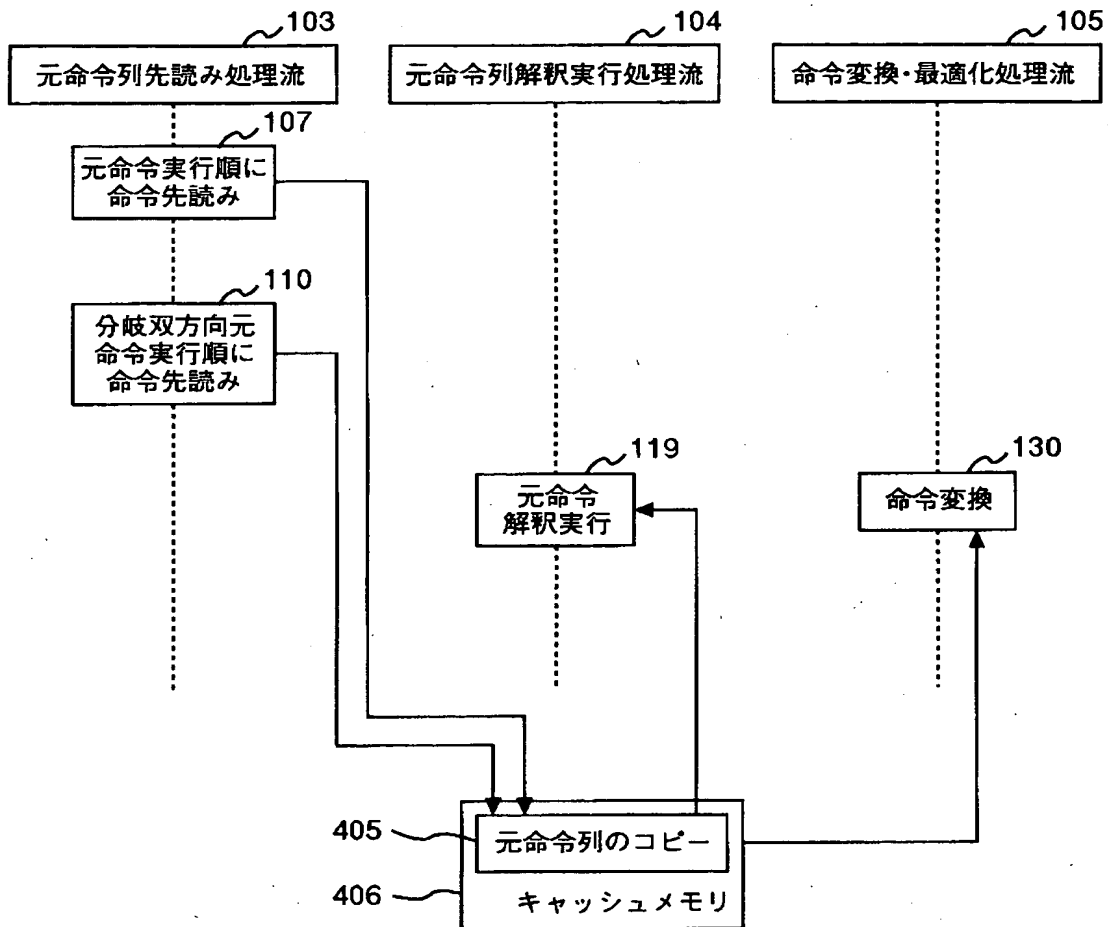
【図 6】

図 6



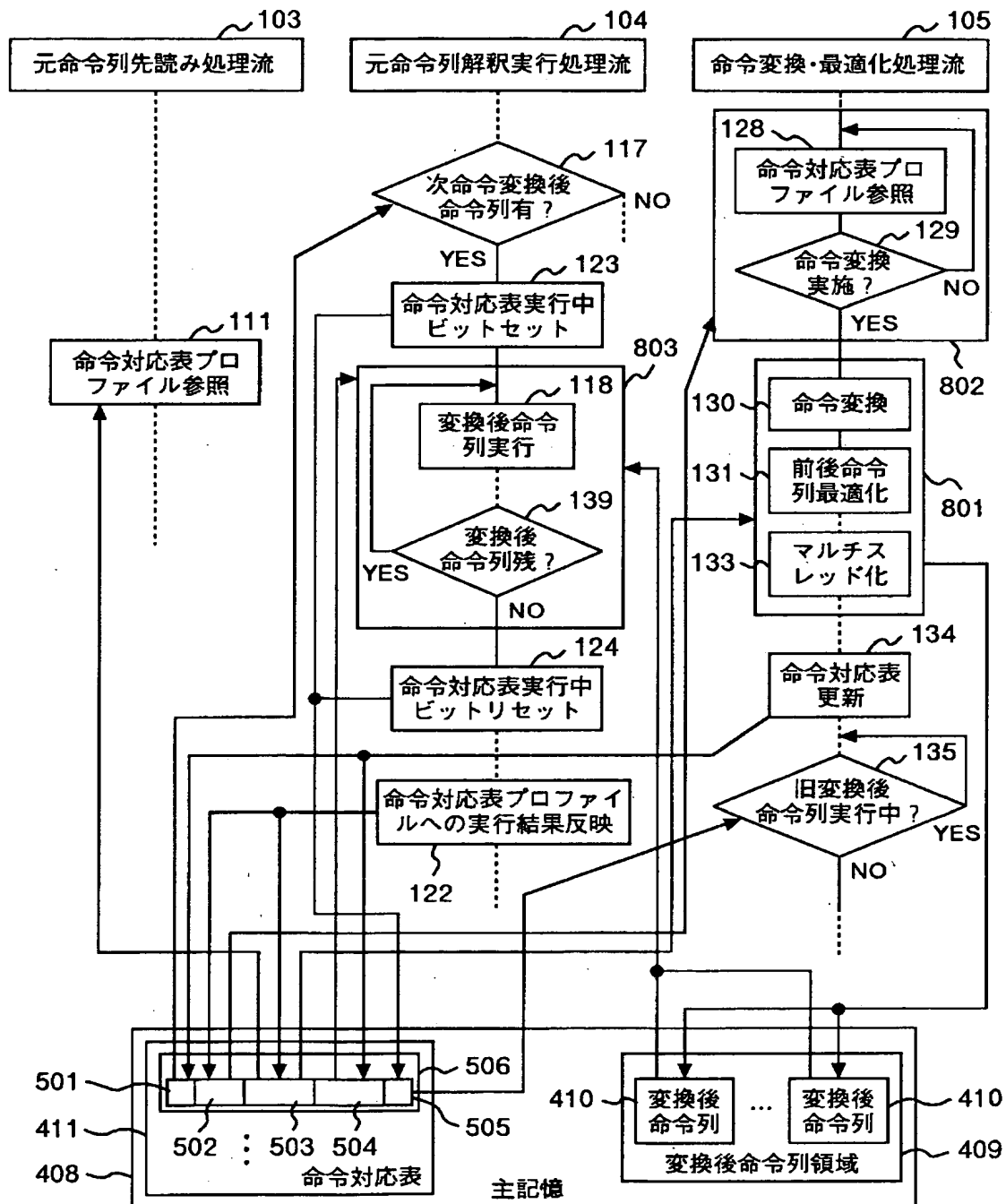
【図 7】

図 7



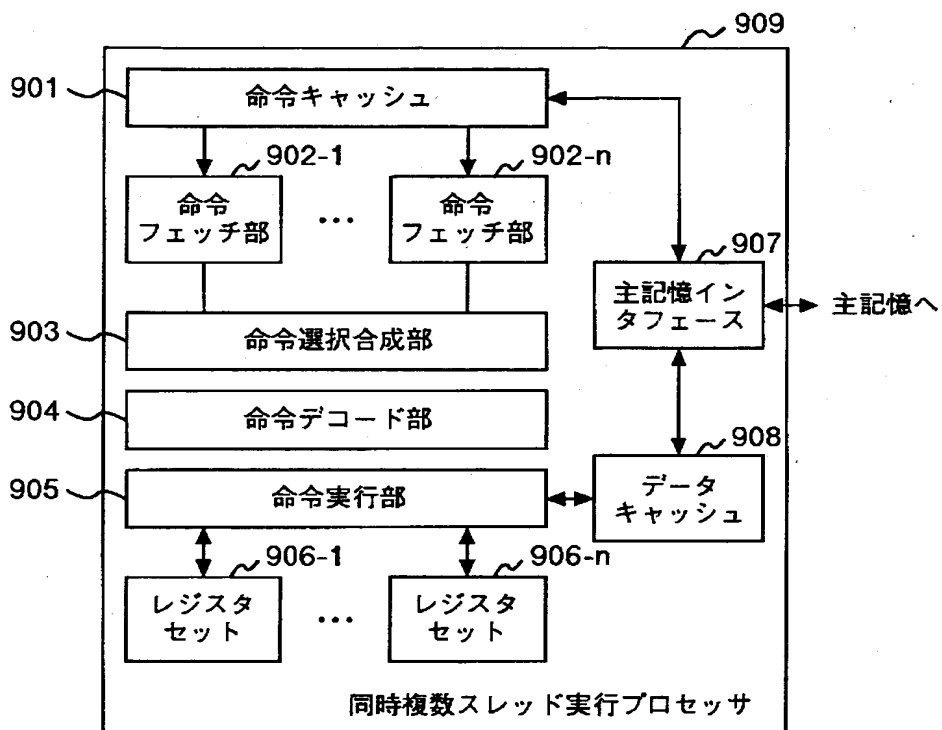
【図 8】

図 8



【図 9】

図 9



【書類名】 要約書

【要約】

【課題】 異種プロセッサ用命令列実行時の命令変換及び最適化処理オーバーヘッドを削減し、同時に、変換後命令列高速処理、プロセッサ高速周波数動作及び低消費電力化を実現する。

【解決手段】 元命令列解釈実行処理流 1 0 4，命令変換・最適化処理流 1 0 5 と元命令列先読み処理流 1 0 3 を独立させ、チップマルチプロセッサ形式や、1 個の命令実行制御部で同時に複数の処理流を実行する形式でプロセッサを構成して上記複数処理流を並列処理する。さらに、命令変換・最適化処理流 1 0 5 は、複数処理流を生成する形で変換後命令列を構成し、元命令列解釈実行処理流 1 0 4 は、各命令解釈実行時に命令変換・最適化処理流 1 0 5 に生成された当該命令に対応する変換後命令列が存在する場合にこれを実行する。

【効果】 上記課題を解決し、さらに元命令列読み込み処理オーバーヘッドを削減する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地
氏 名 株式会社日立製作所